The Fast Lane on the Java Highway

Linux Cluster

Parallel Computing Architecture for Java - Pca4j

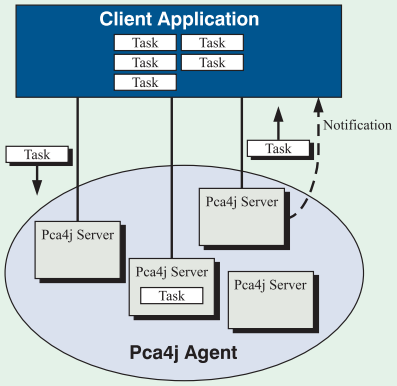
A high performance computing solution

Idaho National Engineering and Environmental Laboratory
INEEL
Home of Science and Engineering Solutions

Pca4j is a framework for Java-based distributed parallel computing. It is designed to integrate easily into Java applications to provide them with parallel computing capability. Client applications that are integrated with Pca4j can access a Pca4jAgent which manages a pool of Pca4jServers that are able to perform tasks for them. The client application must only create a set of tasks, which are Java objects containing the data and code necessary to perform the task. Pca4j then schedules these tasks to available Pca4jServers that are running in a distributed computing environment. When a task is complete, the client is notified and given the results of the task.

Pca4j Objectives

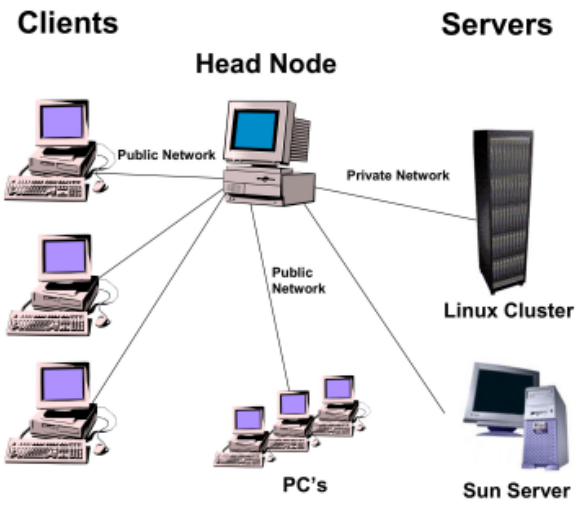
- Provide an easy-to-use framework for parallel computing, facilitating conversion of serial tasks to run in parallel.
- Integrate simply into existing architectures.
- Hide complexity of the distributed computing environment from the user.
- Operate in a cross-platform, heterogeneous environment (Windows, Unix, Linux).
- Work with many types of distributed computing configurations (clusters, SMP machines, grids).



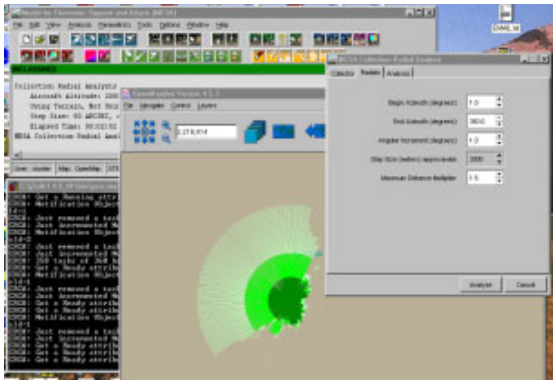
Two Versions

Pca4j comes in two different versions. The standard version is based on the Jini distributed services protocol. This version is a simple, low overhead method for distributing parallel tasks over a distributed computing environment. The primary advantage of this version is the speed that results from low overhead. The advanced version (Pca4jmx) is based on the Java Management Extensions (JMX), and gives the client application more control over scheduling its tasks and communicating with those tasks while they run remotely.

Pca4j is designed to work with a variety of architectures



The Model for Electronics Support and Attack (MESA)



Pca4j and MESA

The Model for Electronic Support and Attack (MESA) is a U.S. Air Force Tactical Decision Aid (TDA) software application. MESA is used to model the capability of a variety of electromagnetic signal entities (radars, jammers, transmitters, receivers, communications equipment, etc). MESA is written in the Java and C++ programming languages. Because of the complexity of modeling electromagnetic wave propagation, it is not uncommon to set up scenarios for analysis that can take many hours or days to complete.

Typical MESA analyses involve computing radials for 360-degree coverage around an electromagnetic entity, one radial for each degree. The calculations for these radials are computationally expensive.

However, as each radial is algorithmically independent from the other radials, MESA is an ideal application for parallelization using the Pca4j framework.

Under Pca4j, each radial is created as a Pca4j task. The tasks are then sent to the Pca4jAgent which distributes them to a set of processors (for example, a linux cluster), each of which are running Pca4j servers. After processing, the results are returned to the client for display.

Impact to the existing MESA software is reasonable. The run-time performance improvement is dramatic. The radials that were the computationally most expensive gained the most by being parallelized. Two examples that illustrate the speedup seen when using a 30-node (60 CPU) Linux cluster are shown in the graph below.

ORIGINAL RUN TIME (SERIAL)	CLUSTER RUN TIME (PARALLEL)	SPEED-UP
6:39	00:14	29 X
6:09:13	6.31	57 X

For more information:

INEEL
Greg Corbett
(208) 526-6295
gtc@inel.gov